МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 4

«Об'єктно-орієнтоване програмування авіаційно-транспортних систем»

Тема: «Реалізація класу і робота з об'єктами»

ХАІ.301.173.3-92ав(і).4ЛР

Виконав студентка гр. \_\_\_\_3-92ав(і)\_\_\_\_\_\_\_

Микитас Анна Олексіївна (П.І.Б.)

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ас.  В. О. Білозерський

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Застосувати теоретичні знання з основ програмування на мові Python з використанням об'єктів і класів, навички використання бібліотеки для візуалізації масивів даних, і навчитися розробляти скрипти для роботи з об'єктами призначених для користувача класів.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Визначити клас Point\_n (n – номер варіанту), який реалізує абстракцію з атрибутами:

1) дві дійсні координати точки на площині (властивості, приховані змінні екземпляра),

− для кожної метод-геттер (повертає відповідну координату),

− для кожної метод-сеттер (записуює відповідну координату, якщо вона у межах [-100, 100], інакше – дорівнює 0))

2) кількість створених екземплярів точки (змінна класу),

3) метод класу (повертає кількість створених примірників),

4) конструктор з двома параметрами (за замовчуванням),

5) деструктор, що виводить відповідне повідомлення,

6) метод, що змінює координати точки з двома вхідними дійсними параметрами:

− зсув по х,

− зсув по у.

Завдання 2. Виконати операції з об'єктами даного класу відповідно до варіанту (див. таб.1).

Завдання 3. Використовуючи пакет matplotlib, відобразити створені об'єкти в графічному вікні до і після змін.

Завдання 4. Зберегти координати точок у текстовому файлі у форматі: номер: координата\_х; координата\_у – для непарних варіантів (номер) координата\_х:координата\_у – для парних варіантів

# ВИКОНАННЯ РОБОТИ

Завдання 1. 3

Клас Point\_3:

Вхідні дані:

x, y: Дійсні координати точки на площині.

Опис: Координати точки на площині.

Тип: float.

Обмеження: -100 <= x, y <= 100.

Вихідні дані:

Немає явно визначених вихідних даних, але клас містить методи для отримання та зміни координат точки.

Діаграму класу

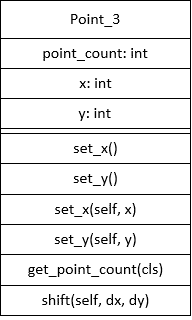


Рисунок 1 – Діаграма класу Point\_3

Лістинг коду вирішення задачі наведено в дод. А (стор. <номер сторінки>).

Завдання 2.

Головна функція main:

Опис роботи функції:

Створення списку з трьох точок класу Point\_3.

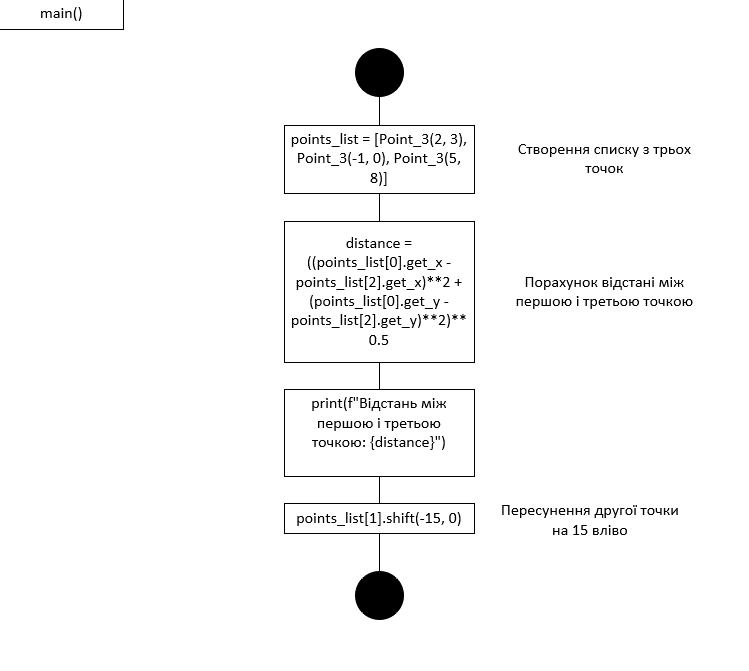
Розрахунок та виведення відстані між першою і третьою точкою.

Пересування другої точки на 15 вліво.

Відображення створених об'єктів до та після змін за допомогою бібліотеки matplotlib.

Збереження координат точок у текстовому файлі.

Алгоритм показано нижче



Лістинг коду вирішення задачі наведено в дод. А (стор. <номер сторінки>). Екран роботи програми показаний на рис. Б.1.

Завдання 3.

Відображення точок до та після змін:

Опис роботи фрагменту коду: Створення списку точок та їх відображення до змін.

Відображення точок після змін.

Алгоритм вирішення показано нижче

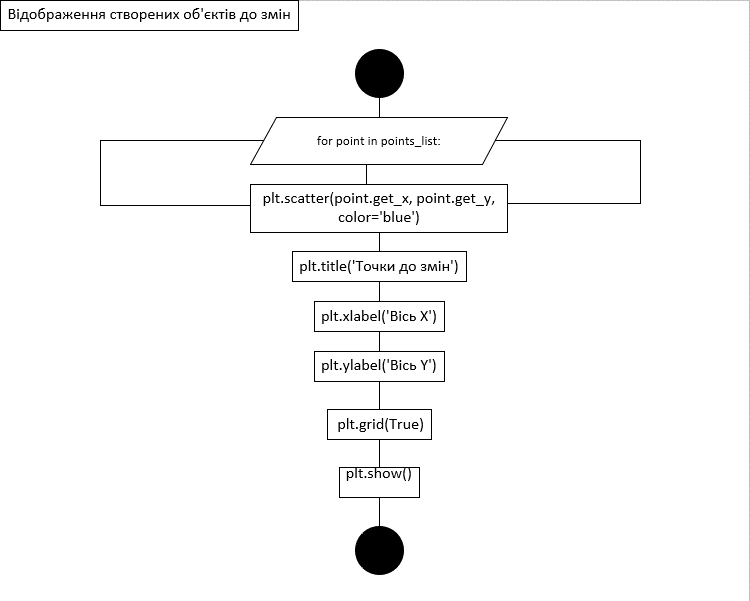


Рисунок 3 – Відображення створених об’єктів до змін

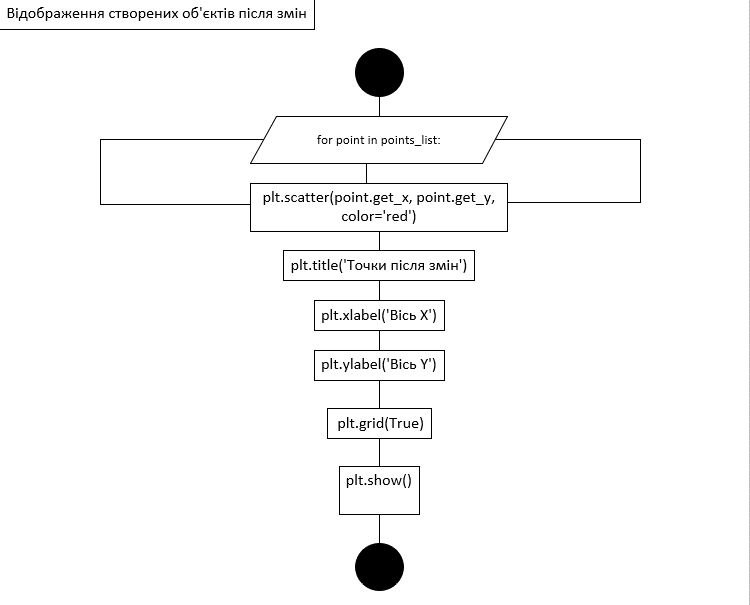


Рисунок 4 – Відображення створених об’єктів після змін

Лістинг коду вирішення задачі наведено в дод. А (стор. <номер сторінки>). Екран роботи програми показаний на рис. Б.2-3.

Завдання 4.

Збереження координат у файл:

Вхідні дані:

file\_path: Шлях до текстового файлу.

Опис: Шлях до файлу, у якому будуть збережені координати точок.

Тип: str.

Обмеження: Немає обмежень.

Вихідні дані:

Немає явно визначених вихідних даних, але координати точок зберігаються у файлі.

Алгоритм вирішення показано нижче

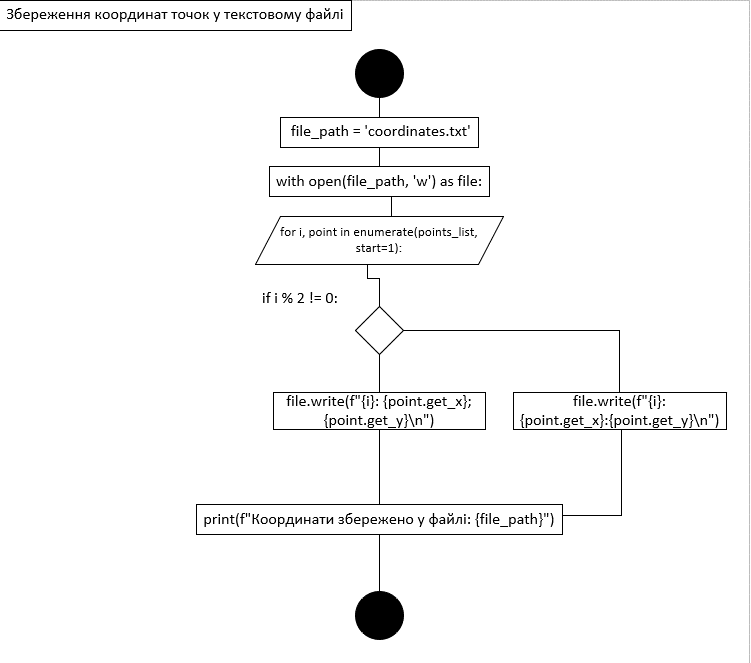


Рисунок 1 – Збереження координат точок у текстовому файлі

file\_path = 'coordinates.txt'

Лістинг коду вирішення задачі наведено в дод. А (стор. <номер сторінки>). Екран роботи програми показаний на рис. Б.4-5.

# ВИСНОВКИ

У ході виконання лабораторної роботи з реалізації класу та роботи з об'єктами в Python, учасники були ознайомлені з концепцією класів та об'єктно-орієнтованого програмування. Вони успішно вивчили створення класу Point\_3, забезпечили обмеження координат точок у межах [-100, 100], та вивчили роботу з методами класу, такими як shift, що дозволяє змінювати координати точки. Учасники виконали операції з об'єктами класу, такі як розрахунок відстані між точками та їх пересування. Крім того, вони закріпили отримані знання про роботу з бібліотекою matplotlib, відображаючи точки на графіку перед та після змін. Навички збереження координат у текстовому файлі були успішно відпрацьовані, дозволяючи учасникам ефективно взаємодіяти з даними.

ДОДАТОК А

Лістинг коду програми до задач 1-4

import matplotlib.pyplot as plt

class Point\_3:

    # Кількість створених екземплярів точки

    point\_count = 0

    def \_\_init\_\_(self, x=0.0, y=0.0):

        # Дійсні координати точки на площині

        self.\_\_x = 0.0

        self.\_\_y = 0.0

        # Задаємо координати точки через метод, щоб використати обмеження [-100, 100]

        self.set\_x(x)

        self.set\_y(y)

        # Збільшуємо кількість створених екземплярів точки

        Point\_3.point\_count += 1

    def \_\_del\_\_(self):

        # Зменшуємо кількість створених екземплярів точки при видаленні

        Point\_3.point\_count -= 1

        print("Point\_3 object destroyed!")

    @property

    def get\_x(self):

        return self.\_\_x

    @property

    def get\_y(self):

        return self.\_\_y

    def set\_x(self, x):

        # Задаємо координату x з обмеженням від -100 до 100

        if -100 <= x <= 100:

            self.\_\_x = x

        else:

            self.\_\_x = 0.0

    def set\_y(self, y):

        # Задаємо координату y з обмеженням від -100 до 100

        if -100 <= y <= 100:

            self.\_\_y = y

        else:

            self.\_\_y = 0.0

    @classmethod

    def get\_point\_count(cls):

        # Повертає кількість створених екземплярів точки

        return cls.point\_count

    def shift(self, dx, dy):

        # Змінює координати точки на dx по x і dy по y

        self.set\_x(self.\_\_x + dx)

        self.set\_y(self.\_\_y + dy)

def main():

    # Задача 2: Виконання операцій з об'єктами класу

    # Створення списку з трьох точок

    points\_list = [Point\_3(2, 3), Point\_3(-1, 0), Point\_3(5, 8)]

    # Порахунок відстані між першою і третьою точкою

    distance = ((points\_list[0].get\_x - points\_list[2].get\_x)\*\*2 + (points\_list[0].get\_y - points\_list[2].get\_y)\*\*2)\*\*0.5

    print(f"Відстань між першою і третьою точкою: {distance}")

    # Пересунення другої точки на 15 вліво

    points\_list[1].shift(-15, 0)

    # Задача 3: Використання бібліотеки matplotlib

    # Відображення створених об'єктів до змін

    for point in points\_list:

        plt.scatter(point.get\_x, point.get\_y, color='blue')

    plt.title('Точки до змін')

    plt.xlabel('Вісь X')

    plt.ylabel('Вісь Y')

    plt.grid(True)

    plt.show()

    # Відображення створених об'єктів після змін

    for point in points\_list:

        plt.scatter(point.get\_x, point.get\_y, color='red')

    plt.title('Точки після змін')

    plt.xlabel('Вісь X')

    plt.ylabel('Вісь Y')

    plt.grid(True)

    plt.show()

    # Задача 4: Збереження координат точок у текстовому файлі

    file\_path = 'coordinates.txt'

    with open(file\_path, 'w') as file:

        for i, point in enumerate(points\_list, start=1):

            if i % 2 != 0:

                file.write(f"{i}: {point.get\_x}; {point.get\_y}\n")

            else:

                file.write(f"{i}: {point.get\_x}:{point.get\_y}\n")

    print(f"Координати збережено у файлі: {file\_path}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

ДОДАТОК Б

Скрін-шоти вікна виконання програми



Рисунок Б.1 – Екран виконання програми для вирішення завдання   
2

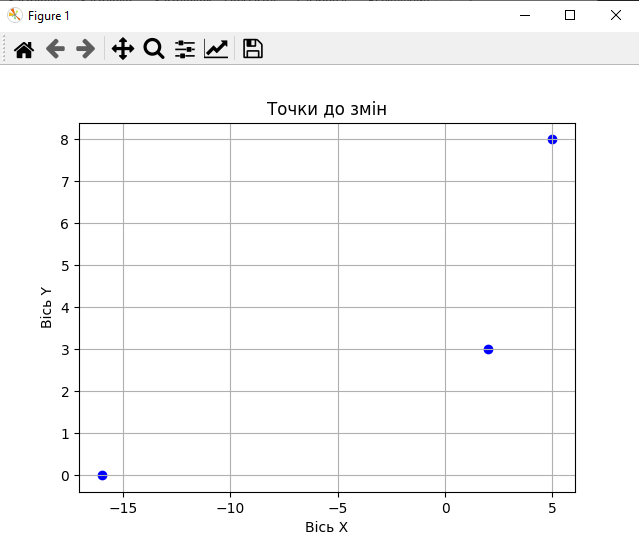


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
3

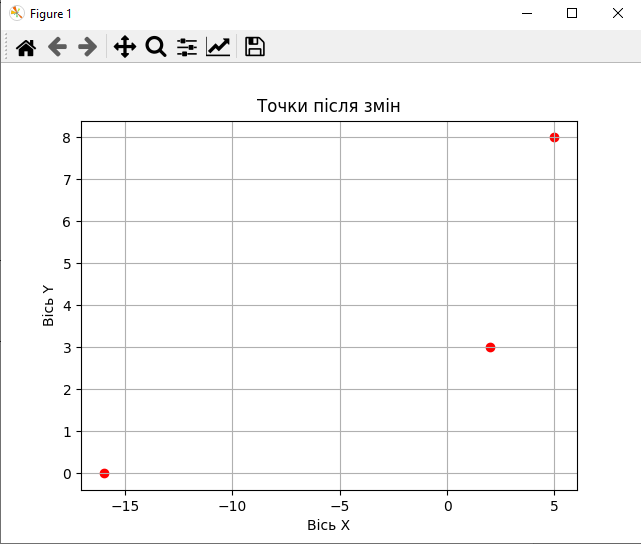


Рисунок Б.3 – Екран виконання програми для вирішення завдання   
2

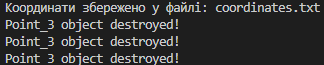


Рисунок Б.4 – Екран виконання програми для вирішення завдання   
4

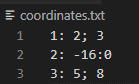


Рисунок Б.5 – Екран виконання програми для вирішення завдання   
4